TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2010

MID-YEAR

COMP 421

Machine Learning

Time Allowed: 3 Hours

Instructions: Answer FOUR questions (180 marks).

There are FIVE questions to choose from: each question is worth 45 marks.

If you answer more questions, only your best 4 will be taken.

Pay close attention to the number of marks for each sub-question, which gives an indication of the depth of answer that is expected.

Non-electronic Foreign-English language dictionaries are permitted.

Question 1. [45 marks]

Define the "bentness" b of a coin to be the fraction of tosses that would come up HEADS if it were tossed an infinite number of times. Suppose that the following data $\mathcal{D}$ has been observed: TAIL, TAIL, TAIL, HEAD.

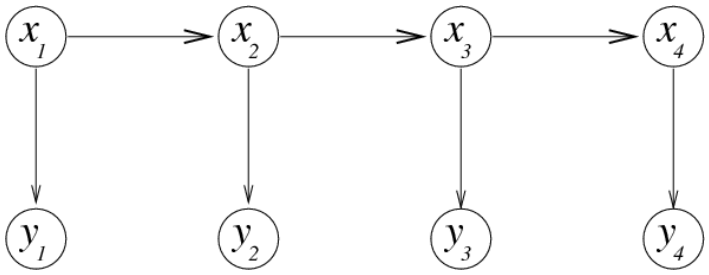
(a) [2 marks]
What is the *maximum likelihood* value of b?

(b) [5 marks]
Assuming a flat prior $p(b)$, sketch the posterior distribution $p(b|\mathcal{D})$, for this data.

(c) [4 marks]
What would the *maximum a posteriori (MAP)* value be for the bentness if you had instead used a prior $p(b) \propto b^2$? Explain your reasoning. (*Hint: you can answer this without needing to do any differentiation*).

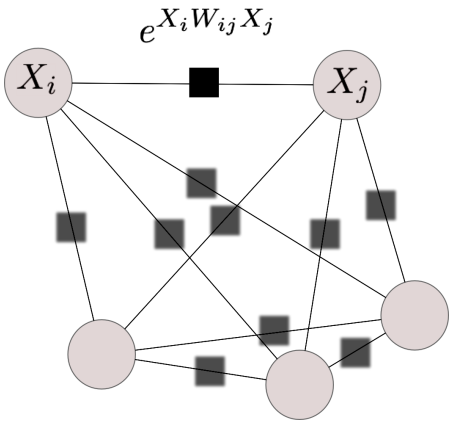
Consider the usual HMM (hidden Markov model) graphical model, shown below. This models temporal data (y) that comes from a single cause (x) over time. But a more realistic model would be able to represent *lots* of such causes.

(d) [5 marks]
Draw an enhanced HMM capable of modelling multiple, independent, temporal causes.



(e) [5 marks]
Such a model raises issues for inference however. What are the issues?

Consider the figure below, which shows the undirected factor graph corresponding to the Boltzmann machine.



Gibbs Sampling in this graphical model amounts to choosing random units and updating their binary state using the sigmoid "neuron" function, familiar from neural networks.

(f) [4 marks]
Why do we require that self-weights W_{ii} be zero in the Boltzmann machine?

(g) [4 marks]
In what sense is an RBM (restricted Boltzmann machine) equivalent to a deep belief net?

(h) [8 marks]
In COMP421 we looked at a way of using Gaussian process (GP) predictions to maximize an unknown function (i.e. optimization), which might be a good idea when data is expensive to obtain. The algorithm used the GP predictive distribution to estimate the *expected improvement* to be obtained from evaluating the unknown function at any given point in the search space, and it adapted the GP's hyperparameters as the search progressed. Could this algorithm exhibit emergent "momentum" in its pattern of search? If possible give a simple 1-D example.

(i) [8 marks]
One idea for training a deep belief net of sigmoid units is the following "greedy" layer-by-layer procedure. First we train the "bottom" layer of weights alone, and freeze those weights. The next layer can then be trained using a new training set produced by sampling from the posterior over hidden units in the first layer, and freezing these weights once they are trained. This can then be continued for successive layers in the network.
Give an explanation for why this procedure does *not* work well in practice.

Question 2. [45 marks]

(a) [3 marks]
What is meant by "detailed balance" in Markov Chain Monte Carlo (MCMC)?

(b) [3 marks]
Why is it easy to *sample* an x from $P(x)$ in a belief net?

(c) [3 marks]
Why is it easy to *evaluate* $P(x)$ for known x in a belief net?

(d) [3 marks]
Why is it hard to *sample* from $P(h|v)$ in a belief net?

(e) [8 marks]
In lectures we discussed the use of Gibbs Sampling for drawing samples in sigmoid belief networks. Discuss how one might use the Metropolis algorithm instead for the same purpose.

(f) [8 marks]
Two modifications to the original Boltzmann machine have enabled restricted Boltzmann machines (RBMs) to be used in practical settings. Outline these modifications.

(g) [5 marks]
The SARSA learning algorithm uses the following learning rule:

$$\Delta Q_{\mathbf{s}_t,a_t} = \eta \ \left[r_{t+1} + \gamma Q_{\mathbf{s}_{t+1},a_{t+1}} - Q_{\mathbf{s}_t,a_t}\right]$$

whereas the algorithm known as "Q-learning" uses this instead:

$$\Delta Q_{\mathbf{s}_t,a_t} = \eta \ \left[r_{t+1} + \gamma \max_{a'} Q_{\mathbf{s}_{t+1},a'} - Q_{\mathbf{s}_t,a_t}\right]$$

What is the significance of the difference between these two rules?

Consider a robot arm consisting of several segments, with actuators that determine the joint angles, $\mathbf{x}$, and sensors that indicate the absolute location of the end of the arm, $\mathbf{y}$. We would like to be able to give the system a desired location $\mathbf{y}^{\text{des}}$, and have it generate appropriate actions to take it there. Suppose we build a data set consisting of examples of actions (the joint angles) $\mathbf{x}$ together with the resulting locations $\mathbf{y}$.

One way to generate sensible actions given a target $\mathbf{y}^{\text{des}}$ would be to use a 'nearest neighbour' approach: simply look up the $\mathbf{y}$ in the dataset that is closest to $\mathbf{y}^{\text{des}}$, and carry out the corresponding action $\mathbf{x}$. This solution requires storing all the data however, and we might prefer a feed-forward neural network to implement the mapping: we could train it with $\mathbf{y}$ vectors as its 'input' and $\mathbf{x}$ vectors as its 'output'. This solution suffers from a major drawback in that the mapping from actions $\mathbf{x}$ to locations $\mathbf{y}$ may be many-to-one.

(h) [12 marks]
In as much detail as you can (commensurate with the marks for this question), explain why this is a problem and develop an alternative approach that would not suffer from the same drawback.

Question 3. [45 marks]

When training a feed-forward neural network to perform 2-way classification, the following cost function is often used:

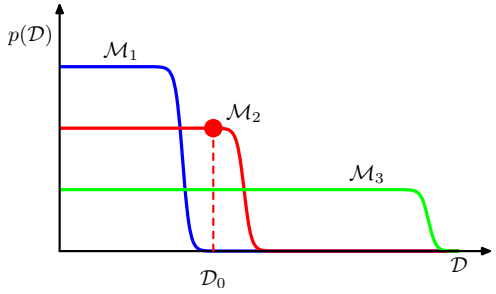
$$C = \sum_n t_n \log y_n \; + \; (1 - t_n) \log(1 - y_n)$$

where n indexes training items, t is the "target" (ie. the training item) and y is the current output of the system, which is real-valued.

(a) [4 marks]
Justify the above cost function in terms of a maximum likelihood / generative model in which y is interpreted as the probability of the system outputting a "1".

(b) [5 marks]
Show that minimizing this cost function leads to the "delta rule" learning rule if applied to a single layer neural network in which y is the sigmoid function of weighted inputs.

Consider the following schematic figure, in which the horizontal axis represents the space of possible data sets. The vertical axis shows $P(\mathcal{D})$, or more correctly $P(\mathcal{D}|\mathcal{M})$, for three different models.



(c) [2 marks]
Which model is the simplest?

(d) [4 marks]
If all three models had similar prior probabilities, which one would you prefer, given the particular data set $\mathcal{D}_0$, and why?

(e) [4 marks]
Rather than choosing one model, a Bayesian might average over all three, weighted by their posterior probabilities. Given that, it might seem that merely adding a 4^{th} model that was very similar to (say) $\mathcal{M}_3$ might significantly change the predictions. What is the flaw in this reasoning?

The next two questions relate to the following factorisation, which is the basis of all belief networks:

$$p(x_1, \ldots, x_K) \;=\; \prod_{i=1}^{K} p(x_i | \text{parents}_i)$$

(f) [6 marks]
In words, state the general independence condition that is assumed by belief nets.

(g) [6 marks]
Demonstate or otherwise argue that the representation for the joint probability encoded by a belief net is correctly normalised, provided each of the conditional probability tables is correctly normalised. (Note: it *might* help to assume that the ordering $x_1, \ldots, x_K$ is an *ancestral ordering*, so that the parents of i are always at indices $j < i$).

(h) [6 marks]
Are Boltzmann machines with *no hidden units* easy to train? Why or why not?

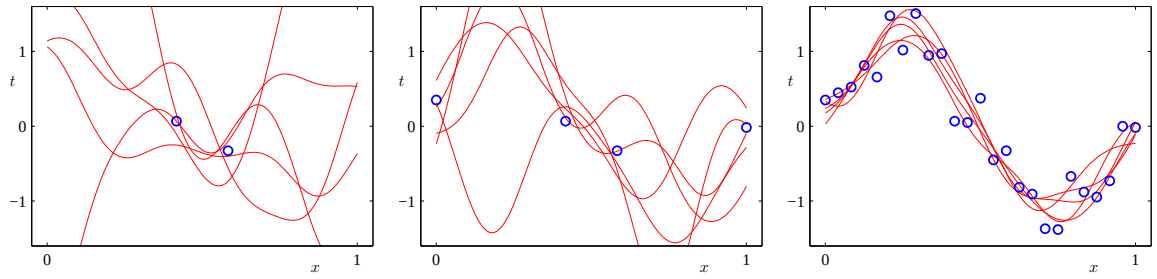
(i) [8 marks]
Function approximators (such as neural networks) can be used to represent the Q-values used in reinforcement learning.
Outline how such a system can learn from experience, including

- what "target" values should the system be trained on?
- what are the pros and cons of using a parameterised function to approximate the Q values?

Question 4. [45 marks]

In supervised learning we have a data set of input-output pairs and a parameterised function $f(\mathbf{x}, \theta)$ that maps inputs to outputs, using parameters θ. One example of such a learner is a neural network in which θ are the weights.

The figures show curves (functions) that are plausible given the data sets (circles): as more data is added, the distribution of plausible curves changes.

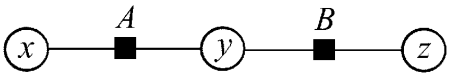


(a) [6 marks]
Describe a method by which you could generate plausible functions like the above.

(b) [10 marks]
Explain the similarities and differences between the SUM-PRODUCT and MAX-SUM algorithms.

(c) [6 marks]
Explain the connection between weight decay (a heuristic used for complexity control in neural nets) and inference involving a prior on the weights.

(d) [3 marks]
Consider the undirected graphical model:



If this was a directed graphical model, we could interpret the factors as conditional probabilities. Can we interpret the factor A as the probability $p(x, y)$, or similar, in this (undirected) case?

The following equation is known as the Bellman equation for the value function V of the optimal policy:

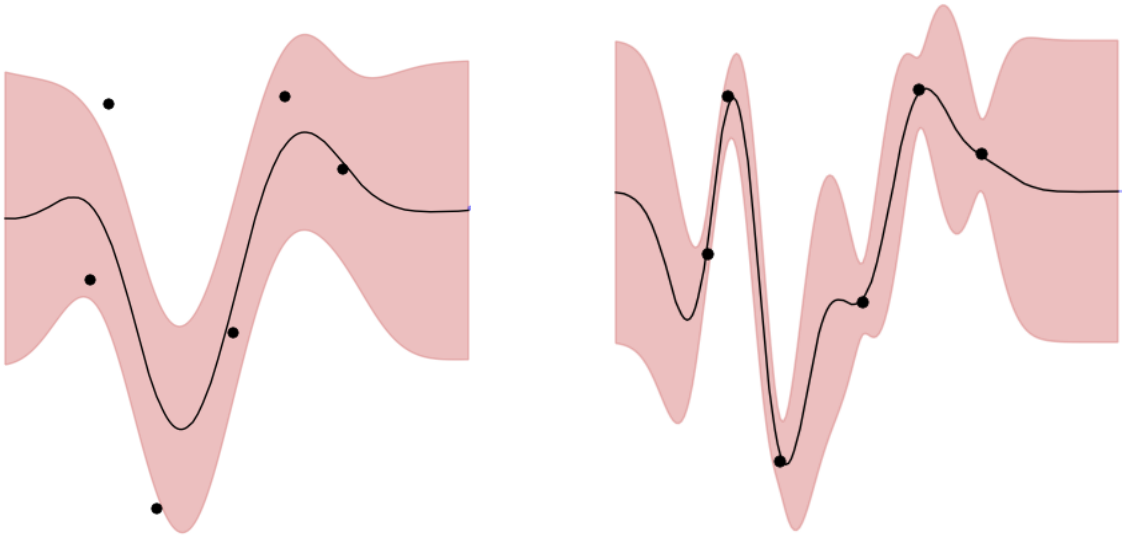
$$V^\star_{\mathbf{s}} = \max_a \sum_{s'} P_{\mathbf{s}'|\mathbf{s},a} \; [R_{\mathbf{s}'} \; + \; \gamma V^\star_{\mathbf{s}'}]$$

Here, $\mathbf{s}$ denotes the current state, a is an action taken in the current state, and $\mathbf{s}'$ is a state at the next time-step. R is the expected immediate reward. The '$\star$' indicates that this is the *optimal* value function.

(e) [4 marks]
Give the algorithm known as 'Value Iteration'.

(f) [8 marks]
Write a similar expression to the Bellman equation above, but for $Q^\star_{s,a}$ instead of $V^\star_{\mathbf{s}}$.

A Gaussian process was "trained" on 6 data points, shown as filled circles in the figures below. The hyperparameters of the Gaussian process were set to random initial values, and then optimized by maximimizing the log likelihood of the data. Depending on the initial random values chosen, this results in either one or the other of the two solutions shown below. Each plot shows the data (circles), the mean of the Gaussian process predictive distribution (solid line) and its error bars (shaded area).



(g) [8 marks]
Explain how it is that the same data can lead to two solutions, and what it is that differs between them.

Question 5. [45 marks]

The joint probability distribution over 4 binary variables (w, x, y, z) can be expressed as a look-up-table with 16 rows, and thus the unconstrained joint $P(w, x, y, z)$ requires 15 free parameters to specify. Suppose you now discover that

- $w \perp\!\!\!\perp y$ (i.e. w and y are independent)
- $w \not\perp\!\!\!\perp y \mid x$ (i.e. w and y are dependent, given x)
- $x \not\perp\!\!\!\perp z$
- $x \perp\!\!\!\perp z|y$

(a) [6 marks]
How many free parameters are required now? Justify your answer.

(b) [6 marks]
Consider random variables a, b, c and d. Show that $a \perp\!\!\!\perp b, c \mid d$ implies that $a \perp\!\!\!\perp b \mid d$.

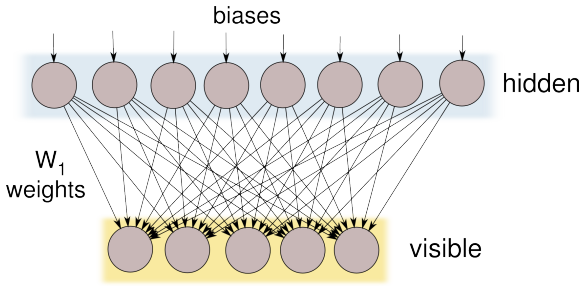
The next three questions concern the use of a generative model for classification. Suppose you are provided with the factors in a generative model having the structure shown below, in which x is a vector and y is a discrete variable.

(c) [4 marks]
Explain how you could use this model to perform classification by inferring y, given a new vector x.

(d) [4 marks]
Explain how you could learn the factors in the above model from a large dataset of $(\mathbf{x}, \mathbf{y})$ pairs.

(e) [8 marks]
Now suppose instead that you have a large dataset of $\mathbf{x}$ examples, but a much smaller number of $(\mathbf{x}, \mathbf{y})$ pairs. How would this affect the learning algorithm?

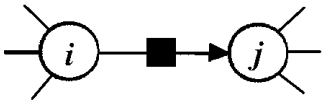
Sigmoid belief nets are stochastic neural nets that can be trained using a data set of vectors describing their "output" nodes. The diagram below shows a sigmoid belief net with one layer of hidden units:



(f) [5 marks]
Give an expression for the likelihood $P(\mathcal{D})$ of a dataset consisting of (say) 100 vectors $\mathbf{v}$: that is, $\mathcal{D} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{100}\}$.

(g) [8 marks]
In a belief network, the messages propagated by the SUM-PRODUCT algorithm correspond to discrete probability distributions over the variables in the graph (some of which may have been observed).

On a diagram like the one shown here, write down the four messages arriving at, and leaving from, the factor denoted by the black square. You can use the short-hand `obs` together with a subscript to denote observations elsewhere in the graph: for example, obs_i denotes all observations in the graph to the left of node i.



(h) [4 marks]
What is the significance of the element-wise *product* of all the messages arriving at node j after the SUM-PRODUCT algorithm has been run?

****